单片机驱动数码管显示设计

关键词：数码管 显示 单片机

摘要：数码管是非常常见的东西，他能显示数字，以及字母，应用非常的广泛，本文和大家谈谈如何用单片机来驱动数码管

数码管是非常常见的东西，他能显示数字，以及字母，应用非常的广泛，本文我来和大家谈谈如何用单片机来驱动数码管

数码管的结构

数码管由7个发光二极管组成,行成一个日字形,它门可以共阴极,也可以共阳极.通过解码电路得到的数码接通相应的发光二极而形成相应的字,这就是它的工作原理.

基本的半导体数码管是由 7 个条状的发光二极管（LED）按图 1 所示排列而成的，可实现数字"0～9"及少量字符的显示。另外为了显示小数点，增加了 1 个点状的发光二极管，因此数码管就由 8 个 LED 组成，我们分别把这些发光二极管命名为"a,b,c,d,e,f,g,dp"，排列顺序如下图 1。

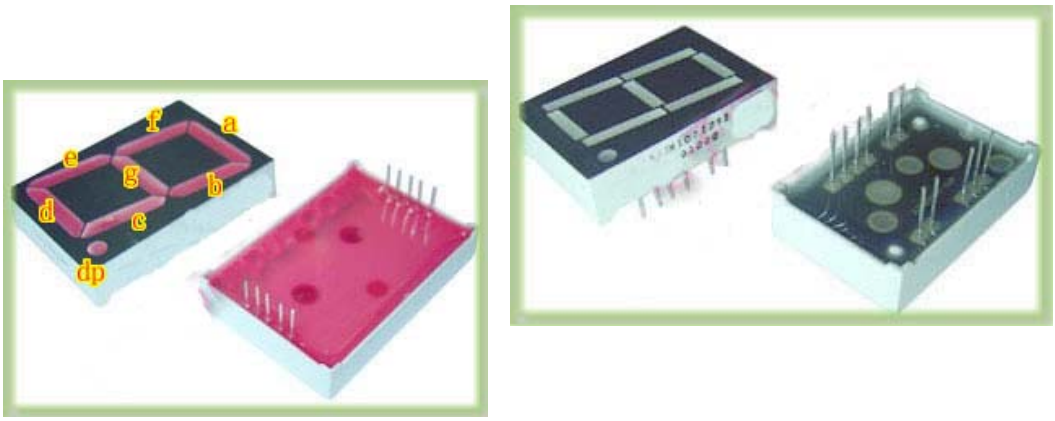

图 1： 数码管引脚图及外形图

数码管的分类

数码管按各发光二极管电极的连接方式分为共阳数码管和共阴数码管两种。

共阴数码管是指将所有发光二极管的阴极接到一起形成公共阴极(COM)的数码管。共阴数码管在应用时应将公共极 COM 接到地线 GND 上，当某一字段发光二极管的阳极为高电平时，相应字段就点亮。当某一字段的阳极为低电平时，相应字段就不亮。共阴数码管内部连接如图 3 所示。

共阳数码管是指将所有发光二极管的阳极接到一起形成公共阳极(COM)的数码管。共阳数码管在应用时应将公共极 COM 接到+5V，当某一字段发光二极管的阴极为低电平时，相应字段就点亮。当某一字段的阴极为高电平时，相应字段就不亮。共阳数码管内部连接如图 2 所示。

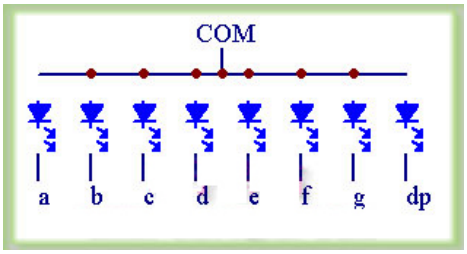

图 2：共阳数码管内部连接图

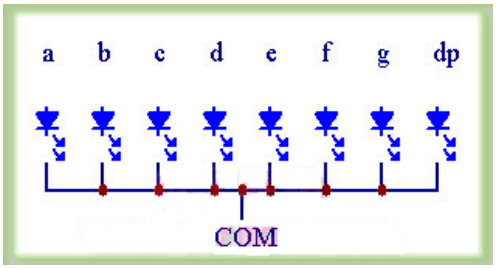

图 3：共阴数码管内部连接图

数码管的显示方式

数码管要正常显示，就要用驱动电路来驱动数码管的各个段码，从而显示出我们要的数字，因此根据数码管的驱动方式的不同，可以分为静态式和动态式两类。

① 动态显示驱动：数码管动态显示接口是单片机中应用最为广泛的一种显示方式之一，动态驱动是将所有数码管的 8 个显示笔划"a,b,c,d,e,f,g,dp"的同名端连在一起，另外为每个数码管的公共极 COM 增加位选通控制电路，位选通由各自独立的 I/O 线控制，当单片机输出字形码时，所有数码管都接收到相同的字形码，但究竟是那个数码管会显示出字形，取决于单片机对位选通 COM 端电路的控制，所以我们只要将需要显示的数码管的选通控制打开，该位就显示出字形，没有选通的数码管就不会亮。通过分时轮流控制各个数码管的的 COM 端，就使各个数码管轮流受控显示，这就是动态驱动。在轮流显示过程中，每位数码管的点亮时间为 1～2ms，由于人的视觉暂留现象及发光二极管的余辉效应，尽管实际上各位数码管并非同时点亮，但只要扫描的速度足够快，给人的印象就是一组稳定的显示数据，不会有闪烁感，动态显示的效果和静态显示是一样的，能够节省大量的 I/O 端口，而且功耗更低。

②静态显示驱动：静态驱动也称直流驱动。静态驱动是指每个数码管的每一个段码都由一个单片机的 I/O 端口进行驱动，或者使用如 BCD 码二-十进制译码器译码进行驱动。静态驱动的优点是编程简单，显示亮度高，缺点是占用 I/O 端口多，如驱动 5 个数码管静态显示则需要 5×8＝40 根 I/O 端口来驱动，要知道一个 89S51 单片机可用的 I/O 端口才 32 个呢：），实际应用时必须增加译码驱动器进行驱动，增加了硬件电路的复杂性。

前面我们学习了数码管的基础知识，现在马上来看看 S51 增强型实验板的数码管吧（图 4），S51 实验板上有 5 位高亮度共阳数码管 DG1～DG5，可以用来做计数器(最大计数值 99999)、温度显示、电子钟等显示实验，掌握数码管的静态显示驱动和动态显示驱动。

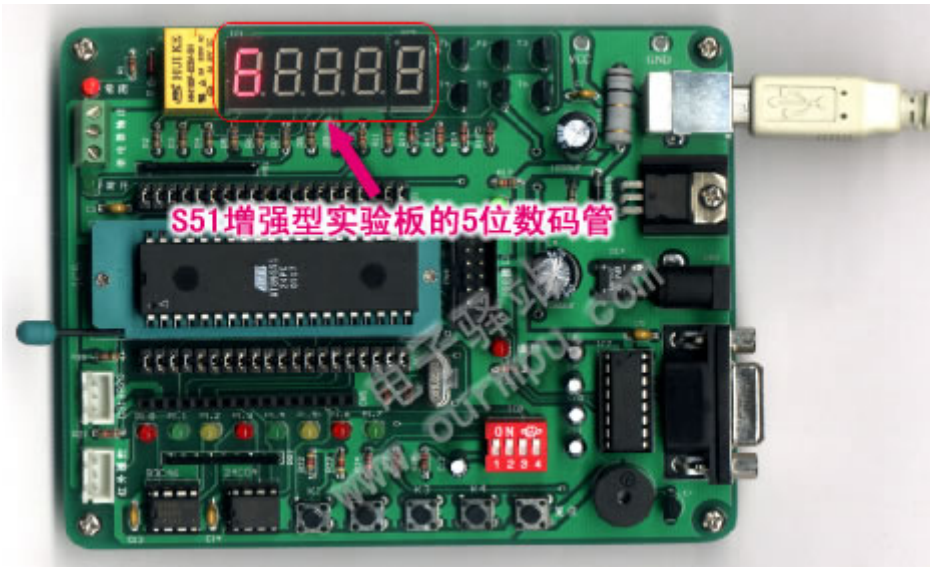

图 4

S51 增强型实验板的数码管驱动原理图如下图 5 所示。

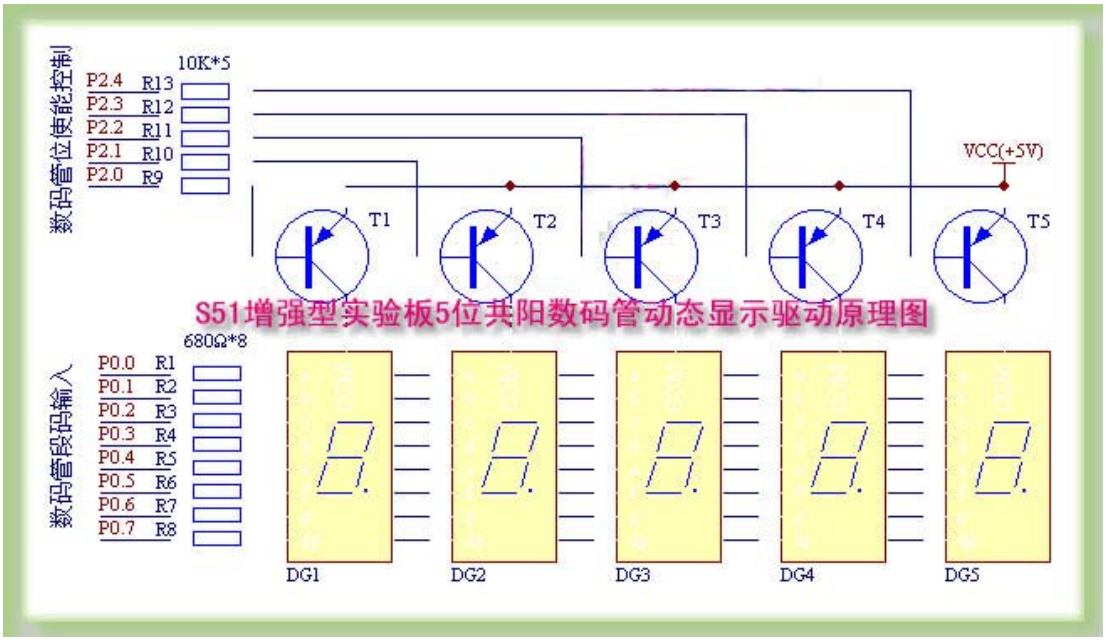

图 5

从原理图可以看出，S51 增强型实验板中数码管的段码 a,b,c,d,e,f,g,dp 分别与单片机的 P0.0～P0.7 相连，控制数码管中显示的字形；数码管的位选通由 5 个 PNP 三极管控制，分别接到单片机的 P2.0、P2.1、P2.2、P2.3、P2.4 端口上，程序中通过控制 P2.0～P2.4 端口的输出电平就可以控制数码管的显示与关闭。如 P2.0 输出低电平时，三极管 T1 导通，+5V 电源加到第一个数码管的 COM 端，那么第一个数码管 DG1 就会显示出相应的数字，显示的数字由单片机 P0.0～P0.7 输出段码决定，当 P2.0 输出高电平时，三极管 T1 截止，数码管 DG1 就不显示，从而实现数码管位选通控制。同理，当 P2.1 输出低电平时，则数码管 DG2 显示。。。

单片机驱动数码管的静态显示编程

上面我们学习了一大堆的理论知识，对数码管已经有了较全面的认识，是否跃跃欲试了呀！马上动手编一个简单的程序验证一下理论吧，现在我们编程让实验板上的第一个数码管 DG1 显示数字“6”，最终的实验效果见上面图 4 所示。

启动 Keil C51 单片机集成开发环境，新建一个工程，工程命名为 smg1.uv2，（不懂如何建立新工程的初学者请看网页 手把手教你建立 Keil 工程），打开一个文本编辑窗，在文本编辑窗中输入如下代码：

```
;**********  下面是数码管显示"6"的程序  **********

MAIN:   CLR      P0.0      ;P0.0 输出低电平,点亮数码管段码"a"
        SETB     P0.1      ;P0.1 输出高电平,熄灭数码管段码"b"
        CLR      P0.2      ;P0.2 输出低电平,点亮数码管段码"c"
        CLR      P0.3      ;P0.3 输出低电平,点亮数码管段码"d"
        CLR      P0.4      ;P0.4 输出低电平,点亮数码管段码"e"
        CLR      P0.5      ;P0.5 输出低电平,点亮数码管段码"f"
        CLR      P0.6      ;P0.6 输出低电平,点亮数码管段码"g"
        SETB     P0.7      ;P0.7 输出高电平,熄灭数码管小数点段码"dp"

        CLR      P2.0      ;P2.0 输出低电平,选通数码管 DG1
```

```
SETB    P2.1    ;P2.1 输出高电平,不选通数码管 DG2
SETB    P2.2    ;P2.2 输出高电平,不选通数码管 DG3
SETB    P2.3    ;P2.3 输出高电平,不选通数码管 DG4
SETB    P2.4    ;P2.4 输出高电平,不选通数码管 DG5

AJMP    MAIN    ;跳转到开始重新进行
END             ;程序结束
```

注：程序中分号“；”后面的中文为每一行程序的注释，是方便我们阅读程序的，可以不输入。

上面的源程序输入完毕后,保存为"smg1.asm"，然后添加到工程 smg1.uv2 中，最后源程序经过编译得到目标文件"smg1.hex"（见下面的图 6 所示），不熟悉 Keil 工程建立、设置及源程序编译等详细操作的初学者请参考网页 >>> 手把手教你建立 Keil 工程。>>>

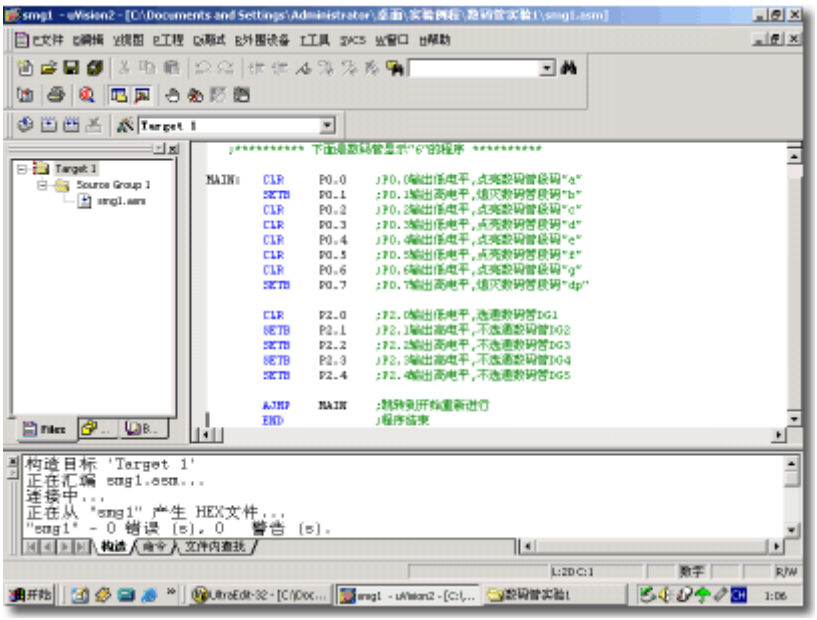

图 6：点击放大该图

现在让我们把目标文件"smg1.hex"烧写到单片机中去，看看实际的效果吧，将ISP 编程器硬件连接好（见下图 7）。

图 7：ISP

将产品配套光盘中的“ISP 编程器驱动软件”文件夹复制到你电脑硬盘的 D 盘根目录下，并将其目录下的所有文件的只读属性去掉，具体操作如下：全选文件夹中的文件，鼠标右键单击出现文件属性对话框，单击“只读”属性前面复选框中的勾，使其只读属性去掉即可。然后双击文件夹中的“ISP 编程器驱动软件.exe”启动编程软件，点击"文件"，在打开文件的对话框中找到工程文件夹中的目标文件"smg1.hex"打开即可，然后点击“AUTORUN”将程序烧写到单片机内部（如下图 8）。

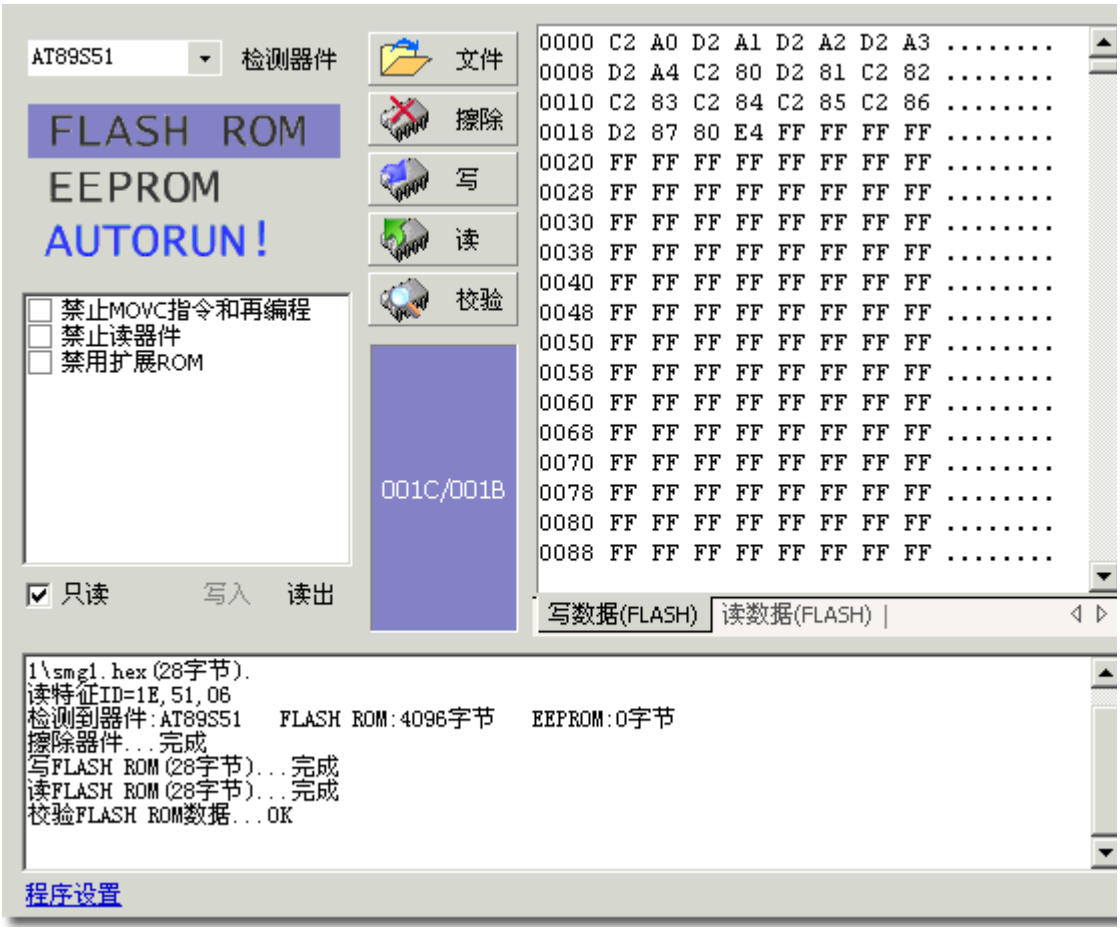

图 8：将程序烧写到单片机内部

烧写完成了，把单片机从编程器中取出，然后插到 S51 增强型实验板上，插上 USB 电源，看看显示效果是不是和上面图 4 显示一样，是否有一点点的成就感呀！初学者也许会问：数码管显示一个数字“6”就要 15 行程序，太复杂了？的确有点复杂了，在上面程序中为了显示数字“6”，数码管的段码"b"、段码"dp"输出的是高电平，其它引脚输出的是低电平，实际上从单片机的 P0.0～P0.7 输出的是二进制码“10000010”，转换成十六进制为 82H。因此，我们只要把所有要显示的数字和字符的段码根据硬件连接编制一个字形表，显示时直接把相应的字形码送到 P0 口就可以了。

共阳 LED 数码管字形(段码)表

显示数字 (字符)	P0.7 "dp"	P0.6 "g"	P0.5 "f"	P0.4 "e"	P0.3 "d"	P0.2 "c"	P0.1 "b"	P0.0 "a"	二进制代码	十六进制代码
0	1	1	0	0	0	0	0	0	11000000	C0H
1	1	1	1	1	1	0	0	1	11111001	F9H
2	1	0	1	0	0	1	0	0	10100100	A4H
3	1	0	1	1	0	0	0	0	10110000	B0H
4	1	0	0	1	1	0	0	1	10011001	99H
5	1	0	0	1	0	1	1	0	10010110	92H
6	1	0	0	0	0	0	1	0	10000010	82H

7	1	1	1	1	1	0	0	0	11111000	F8H
8	1	0	0	0	0	0	0	0	10000000	80H
9	1	0	0	1	0	0	0	0	10010000	90H
A	1	0	0	0	1	0	0	0	10001000	88H
B	1	0	0	0	0	0	1	1	10000011	83H
C	1	1	0	0	0	1	1	0	11000110	C6H
D	1	0	1	0	0	0	0	1	10100001	A1H
E	1	0	0	0	0	1	1	0	10000110	86H
F	1	0	0	0	1	1	1	0	10001110	8EH
H	1	0	0	0	1	0	0	1	10001001	89H

O	1	1	0	0	0	0	0	0	11000000	A3H
P	1	0	0	0	0	1	0	0	10000100	8CH
N	1	1	0	0	1	0	0	0	11001000	C8H

从上面表格中可以看到，显示“6”的十六进制段码值为"82H"，因此我们把刚才的程序修改一下，修改后的完整程序如下：

```
MAIN:   MOV     P0,#82H     ;将数字"6"的段码输出到 P0 口
        MOV     P2,#0FEH    ;从 P2 口输出数码管选通代码，即输出二进制“11111110”
        AJMP    MAIN        ;跳转到开始重新进行
        END                 ;程序结束
```

看看修改后的程序将变得更加简洁，直观了，程序代码从原来的 15 行减少到仅 4 行，一样实现了相同的功能。这也就是我们要学习的编程技巧哦！在编程中尽量用最少的代码实现相同的功能。程序第 1 行的功能是将要显示的数字“6”的十六进制段码"82H"送到 P0 口，程序第 2 行的功能就是将数码管的选通代码#0FEH（即二进制"11111110"）送到 P2 口，从而控制第一位数码管显示，其它数码管熄灭。把修改过的程序编译后的目标文件写到单片机上看到显示效果是一样的。程序中用一行代码 MOV P0,#82H 就输出了字形，因此我们要显示其它字形时只要从上面的数码管段码表中查出对应的十六进制字形码，用同样的方法把段码输出到 P0 口就可以了。比如我们要显示一个数字“8”， 只需将程序中的 MOV P0,#82H 语句改成 MOV P0,#80H 即可，至此，我们终于可以随心所欲地控制数码管要显示的字形了，是不是很简单呀 ：） 。

另外，如果想让第二位数码管 DG2 显示，其它熄灭怎么办呢？其实很简单，只要把对应数码管的选通端口输出低电平就可以使该位数码管显示了，如指令 CLR P2.1 就可以让第二个数码管显示。。。程序中如果使 P2.0～P2.4 都输出低电平，那么实验板上的 5 个数码管都会被选通，显示出相同的字形，即显示“66666”。下面

就是 5 位数码管显示出“66666”的程序，初学者可以实验一下，以加深对数码管显示位选通（使能）控制的理解。

```
MAIN:   MOV     P0,#82H      ;将数字"6"的段码输出到 P0 口
        MOV     P2,#0E0H     ;从 P2 口输出数码管选通代码,使 5 位数码管均选通，即输出二进制“11100000”
        AJMP    MAIN         ;跳转到开始重新进行
        END                  ;程序结束
```

单片机驱动数码管的动态显示编程

上面我们已经学习了数码管静态显示，接下来我们就学习数码管动态显示编程，编程让实验板上的数码管显示“89C51”。从原理图中（图 5）我们可以看到，5 个数码管的 8 个显示笔划"a,b,c,d,e,f,g,dp"的同名端是 2 连在一起的，那么当程序从 P0 口输出字形码时，在同一个时间所有数码管都会接收到相同的字形码。你一定会问，这样不是 5 个数码管都显示相同的数字了吗？如何显示出 5 个不同的字符“89C51”呢？因此，就要使用动态扫描了，在程序中，首先显示一个数，然后关掉；然后显示第二个数，又关掉，显示第三个数，又关掉。。。直到所有要显示的 5 个数完成，再从头开始扫描。轮流点亮扫描过程中，每位显示器的点亮时间是极为短暂的（约 1ms），由于人的视觉暂留现象及发光二极管的余辉效应，尽管实际上各位显示器并非同时点亮，但只要扫描的速度足够快，给人的印象就是一组稳定的显示数据，不会有闪烁感。

数码管显示“89C51”的具体编程思路如下：第一位数码管显示“8” → 延时 1ms → 关闭所有数码管显示 → 第二位数码管显示“9” → 延时 1ms → 关闭所有数码管显示 → 第三位数码管显示“C” → 延时 1ms → 关闭所有数码管显示 → 第四位数码管显示“5” → 延时 1ms → 关闭所有数码管显示 → 第五位数码管显示“1” → 延时 1ms → 关闭所有数码管显示 → 返回到第一步重新进行新一轮扫描过程。下面就是根据该思路编出来的源程序，初学者可以把该程序粘贴到 Keil 工程中，编译得到目标文件，然后烧写到单片机验证一下。

```
; ********** 在数码管上动态显示"89C51" **********
MAIN:   MOV     P0,#80H      ;第 1 位数码管显示“8”
        CLR     P2.0         ;允许第 1 位数码管显示
        ACALL   DELAY        ;显示延时一段时间
```

```
        MOV     P0,#0FFH      ;清除 P0 口字形码
        MOV     P2,#0FFH      ;停止所有数码管显示选通，关闭所有显示


        MOV     P0,#90H       ;第 2 位数码管显示“9”
        CLR     P2.1          ;允许第 2 位数码管显示
        ACALL   DELAY         ;显示延时一段时间
        MOV     P0,#0FFH      ;清除 P0 口字形码
        MOV     P2,#0FFH      ;停止所有数码管显示选通，关闭所有显示

        MOV     P0,#0C6H      ;第 3 位数码管显示“C”
        CLR     P2.2          ;允许第 3 位数码管显示
        ACALL   DELAY         ;显示延时一段时间
        MOV     P0,#0FFH      ;清除 P0 口字形码
        MOV     P2,#0FFH      ;停止所有数码管显示选通，关闭所有显示

        MOV     P0,#92H       ;第 4 位数码管显示“5”
        CLR     P2.3          ;允许第 4 位数码管显示
        ACALL   DELAY         ;显示延时一段时间
        MOV     P0,#0FFH      ;清除 P0 口字形码
        MOV     P2,#0FFH      ;停止所有数码管显示选通，关闭所有显示

        MOV     P0,#0F9H      ;第 5 位数码管显示“1”
        CLR     P2.4          ;允许第 5 位数码管显示
        ACALL   DELAY         ;显示延时一段时间
        MOV     P0,#0FFH      ;清除 P0 口字形码
        MOV     P2,#0FFH      ;停止所有数码管显示选通，关闭所有显示

        AJMP    MAIN          ;跳转到开始重新进行


;******** 延时子程序 ********
    DELAY:  MOV     R1,#10
    Y1:     MOV     R2,#100
            DJNZ    R2,$
```

```
        DJNZ    R1,Y1
        RET


        END
```

【总结】

至此，我们已经较全面地学习了数码管的工作原理和使用方法，相信你对数码管的静态显示、动态显示有了新的认识，掌握了数码管的这两种使用方法，你就可以根据你自己的意愿及要求来编写各种各样的数字显示程序了，如电子温度计、时钟、秒表、频率计、计数器的制作等等，可以充分发挥你的想象达到你所需要的各种显示效果。我们附带的配套软件资料光盘配有相关的实验例程、实验视频录像、单片机多媒体教程、实用电子图书资料、单片机开发软件及编程器、仿真器的全部驱动程序，供大家学习使用，以帮助初学者快速入门。

http://www.icembed.com/info.asp?ArticleID=27717&ArticlePage=1